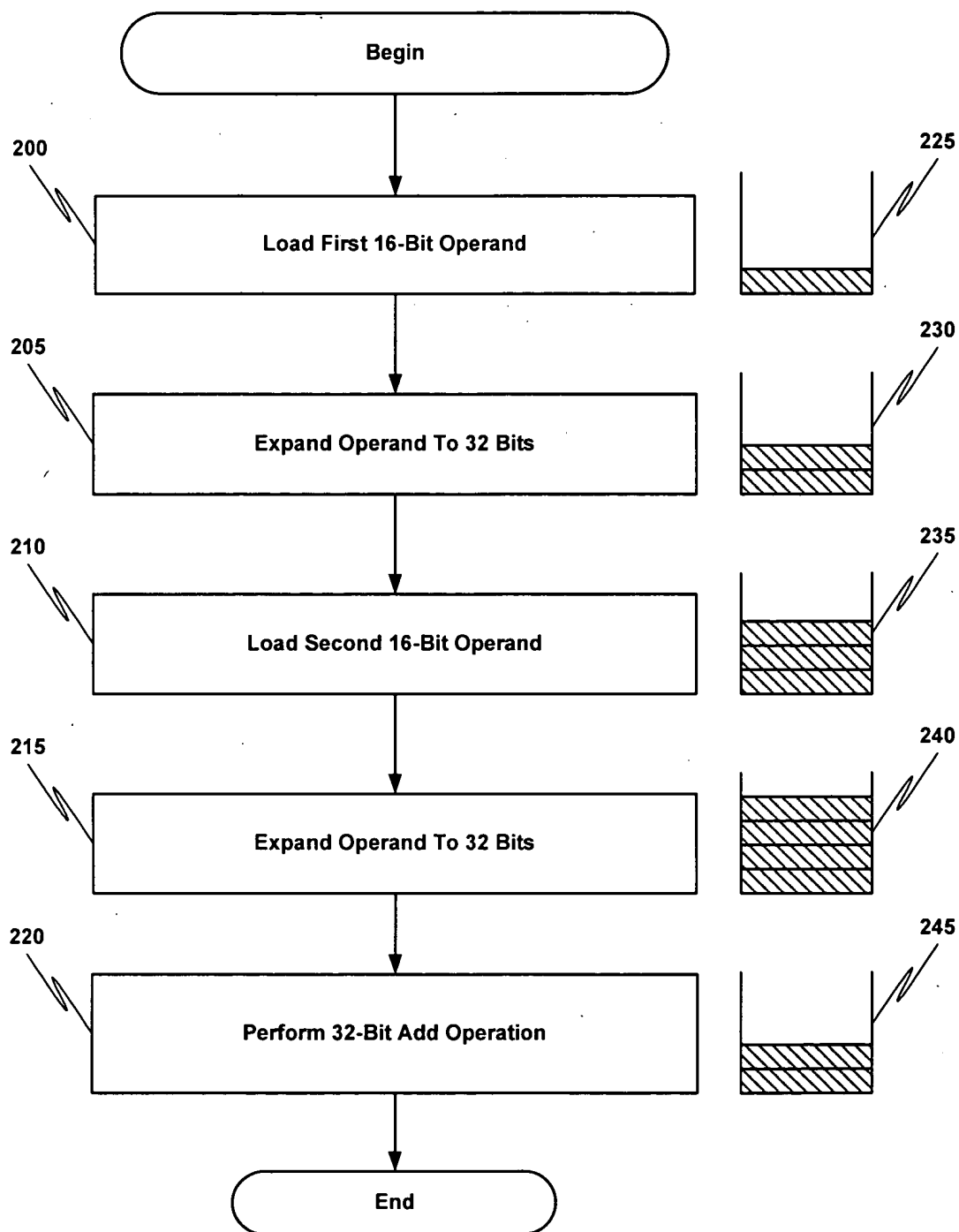


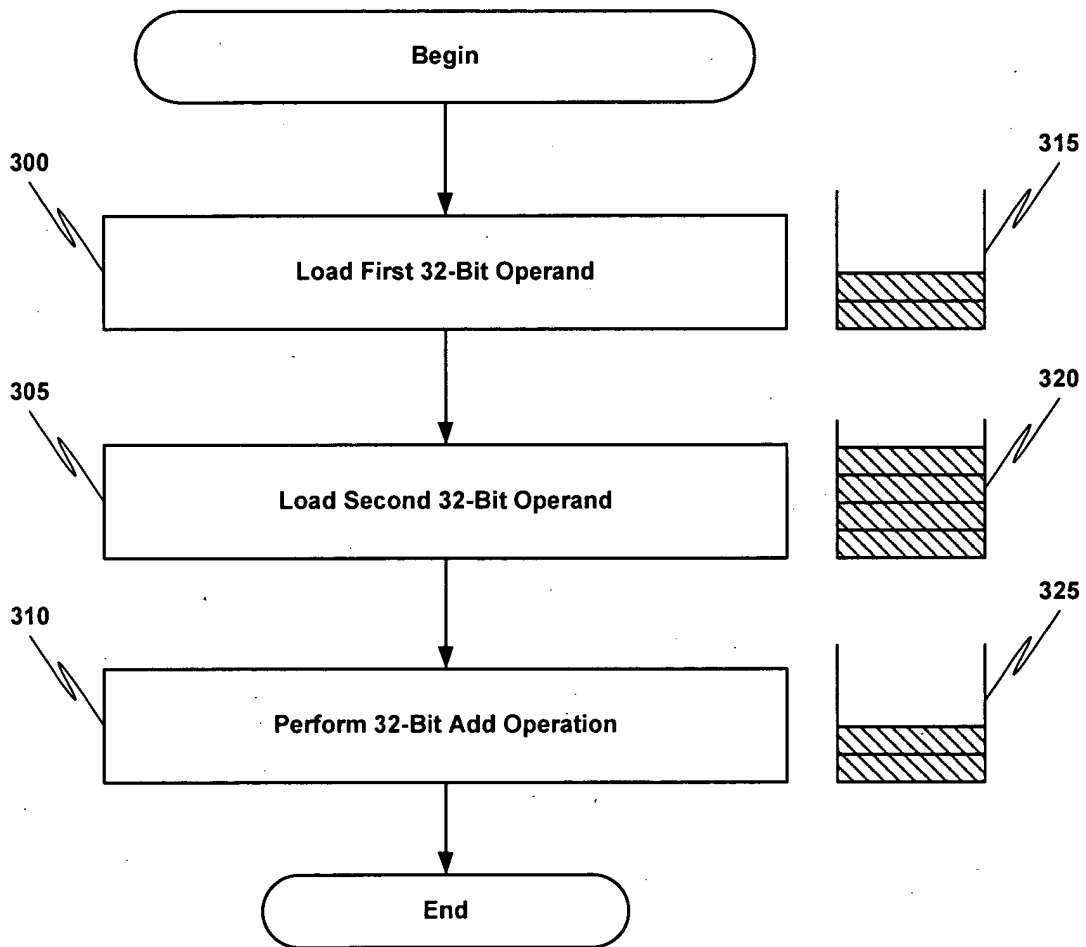
Prior Art

FIG. 1



Prior Art

FIG. 2



Prior Art

FIG. 3

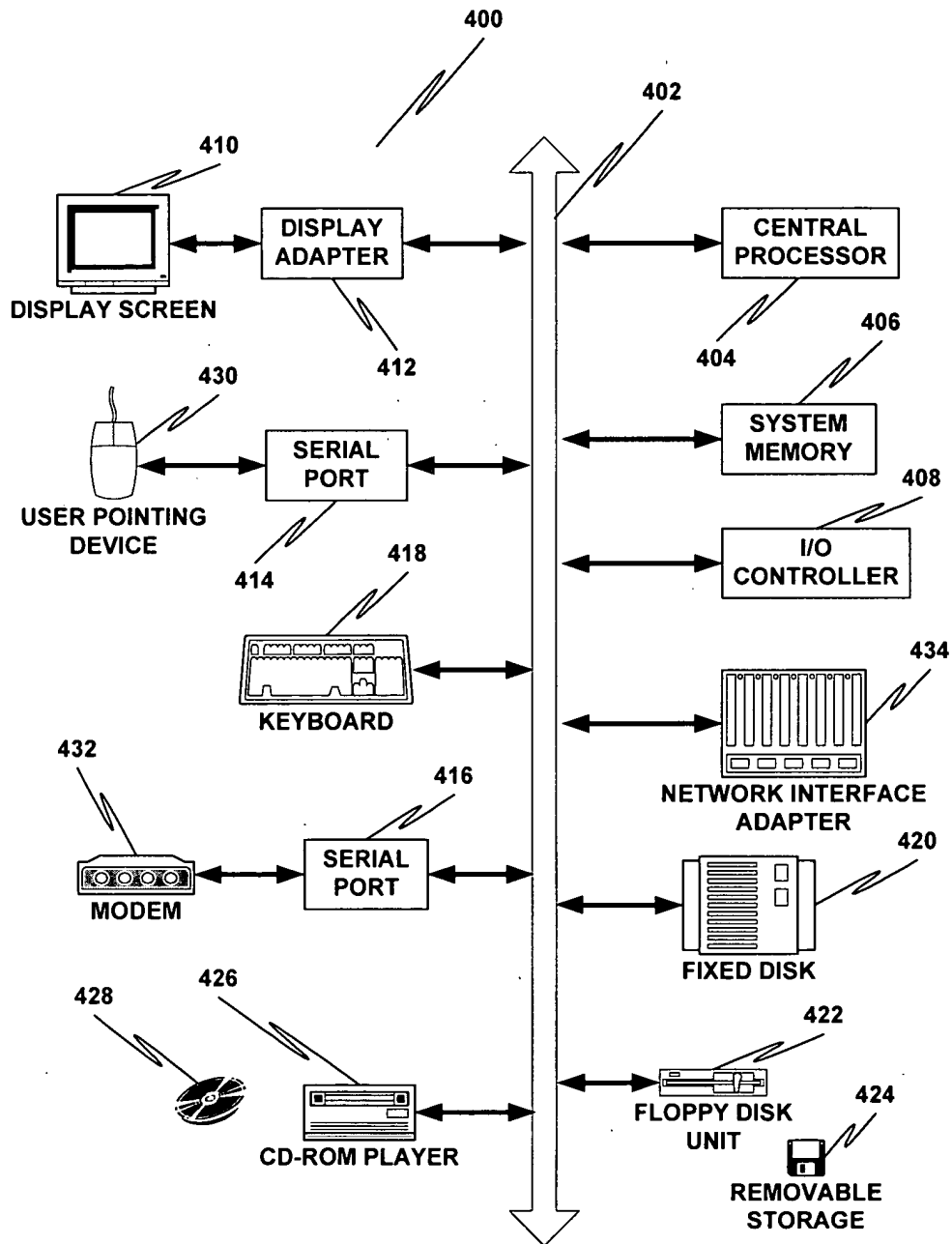


FIG. 4

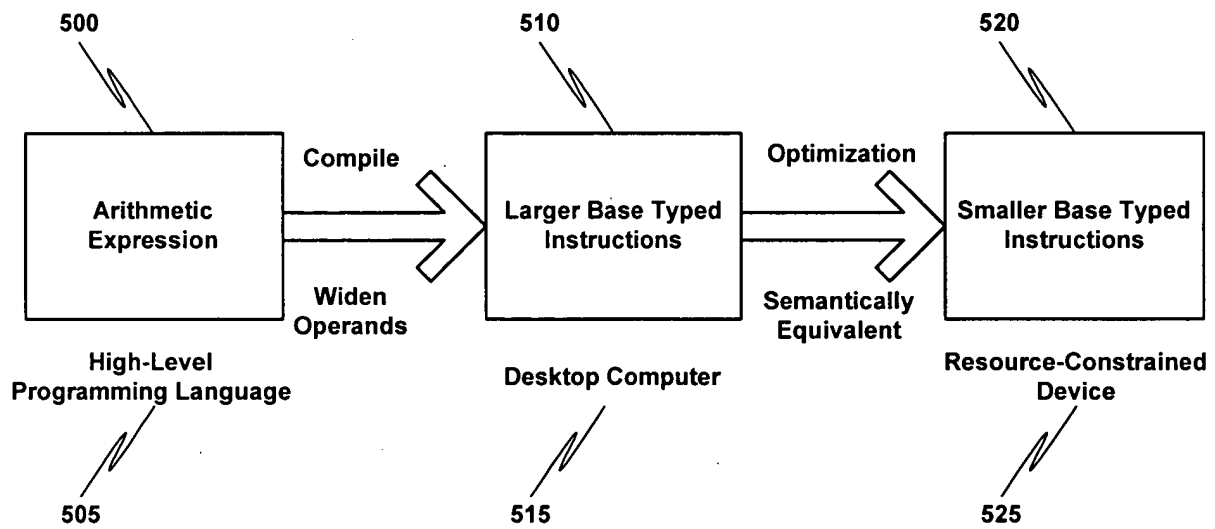


FIG. 5

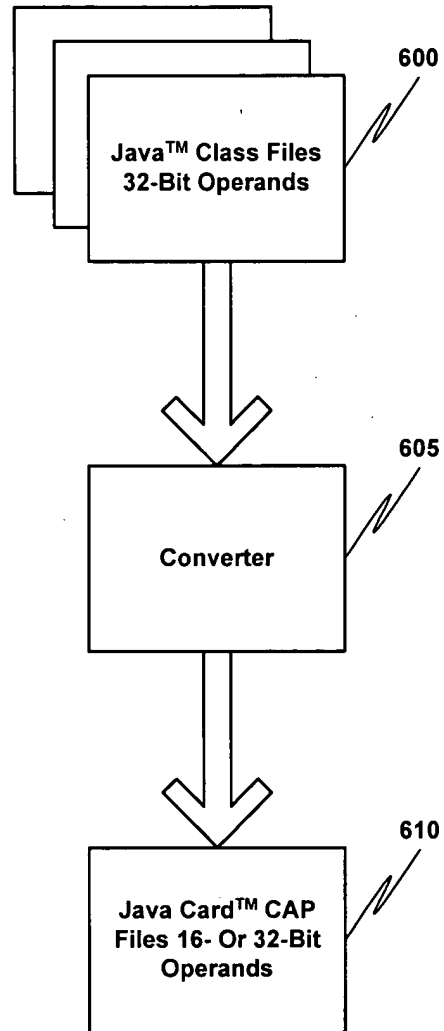


FIG. 6

Desktop Computer

Resource-Constrained Device

```
short a = 0x7FFF;  
short b = 1;  
short c = 1;  
  
result = a [int-type add] b;  
        = 0x00007FFF + 1  
        = 0x00008000  
        = 32768
```

FIG. 7A

```
short a = 0x7FFF;  
short b = 1;  
short c = 1;  
  
result = a [short-type add] b;  
        = 0x7FFF + 1  
        = 0x8000  
        = -32768
```

FIG. 7B

```
result = (short) a [int-type add] b;  
        = (short) (0x00007FFF + 1)  
        = (short) (0x00008000)  
        = 0xFFFF800  
        = -32768
```

FIG. 8A

```
result = (short) a [short-type add] b;  
        = (short) (0x7FFF + 1)  
        = (short) (0x8000)  
        = -32768
```

FIG. 8B

```
result = (short) (a [int-type add] b  
                  [int-type add] c);  
        = (short) (0x00007FFF + 1 + 1)  
        = (short) (0x00008001)  
        = 0xFFFF8001  
        = -32767
```

FIG. 9A

```
result = (short) (a [short-type add] b  
                  [short-type add] c);  
        = (short) (0x7FFF + 1 + 1)  
        = (short) (0x8001)  
        = -32767
```

FIG. 9B

```
result = a[int-type add] b  
        [int-type divide] c;  
        = (0x00007FFF + 1) / 2  
        = 0x00008000 / 2  
        = 0x00004000  
        = 16384
```

FIG. 10A

```
result = a [short-type add] b  
        [short-type divide] c;  
        = (0x7FFF + 1) / 2  
        = 0x8000 / 2  
        = 0xC000  
        = -16384
```

FIG. 10B

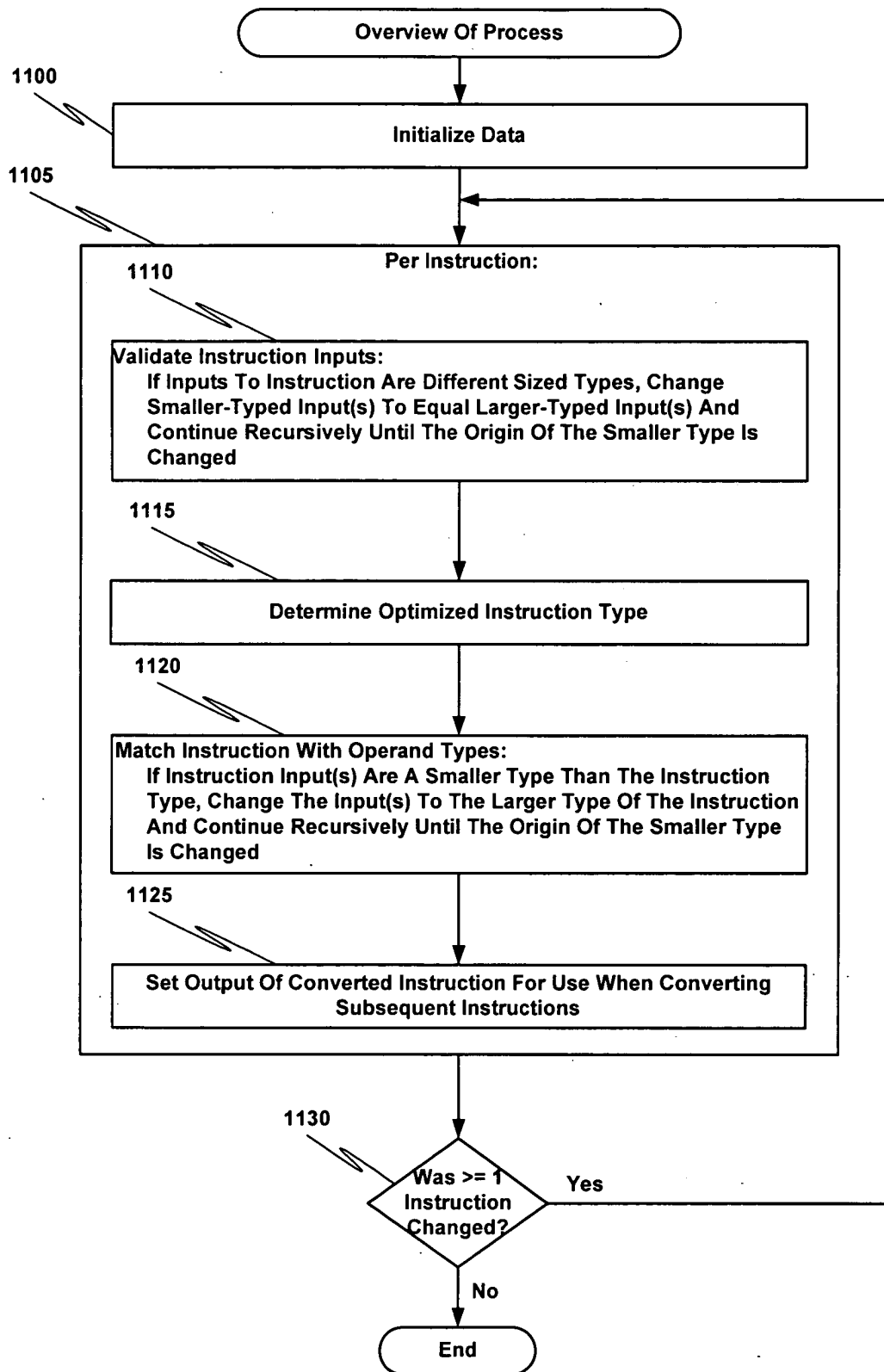


FIG. 11

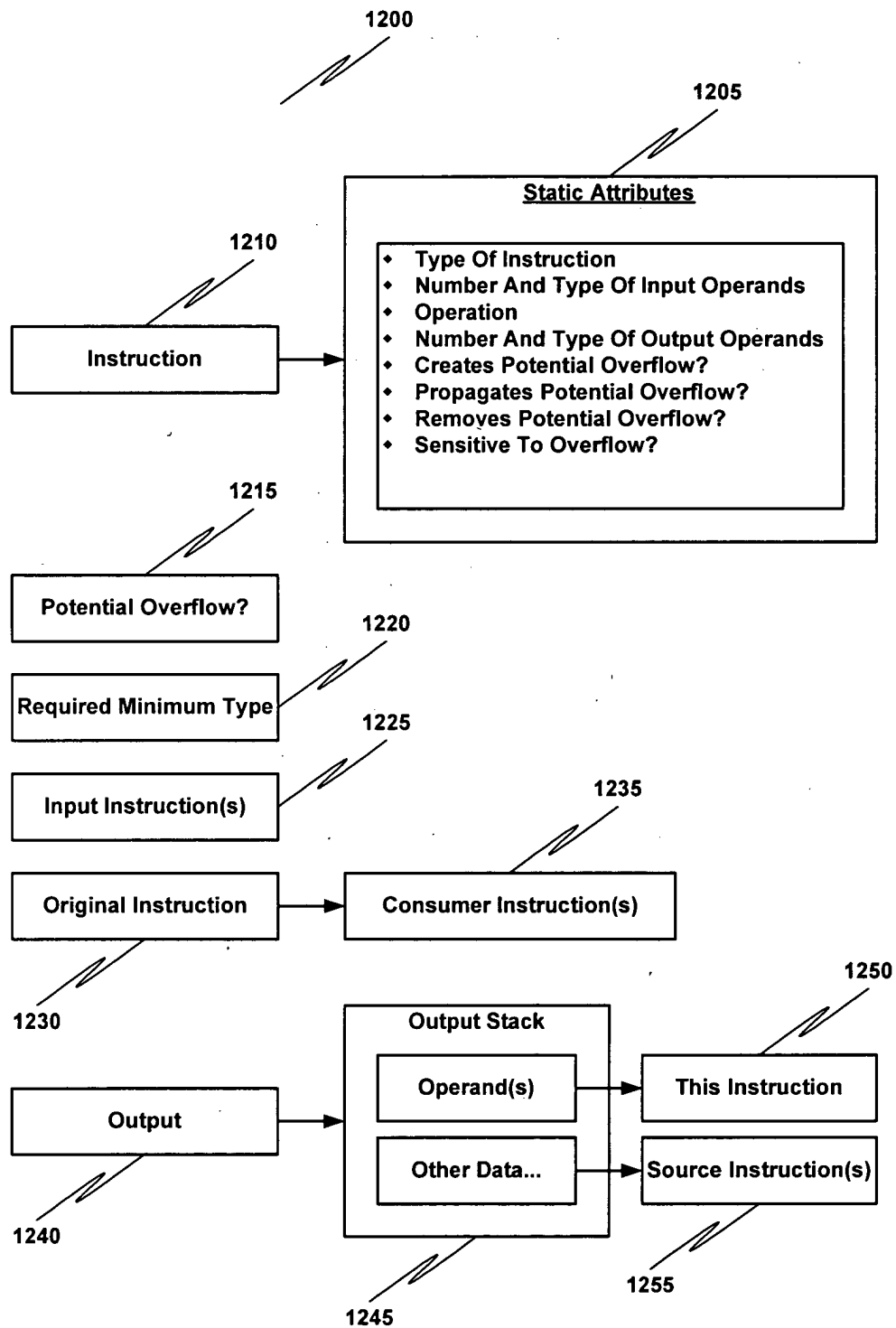


FIG. 12

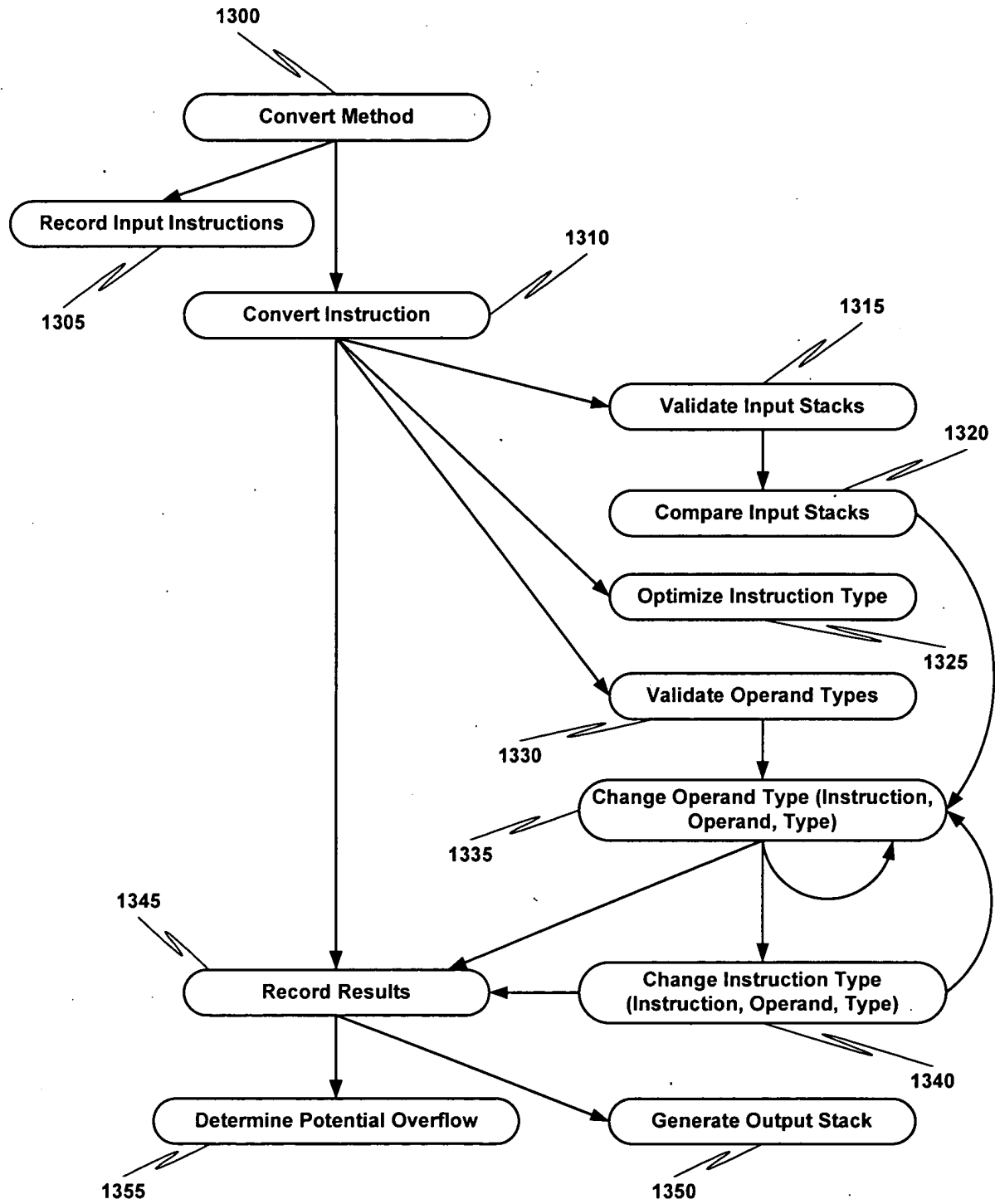


FIG. 13

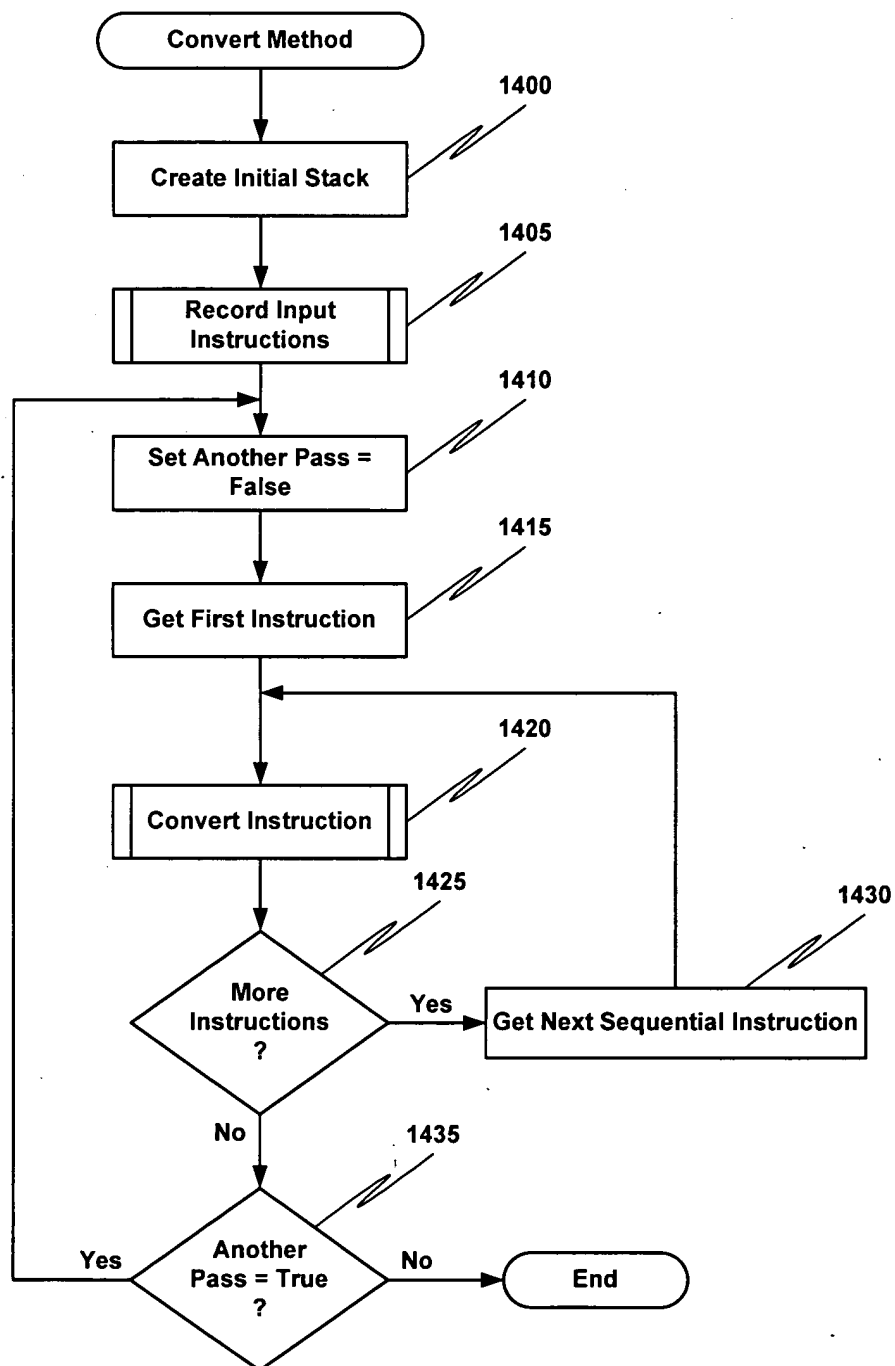


FIG. 14

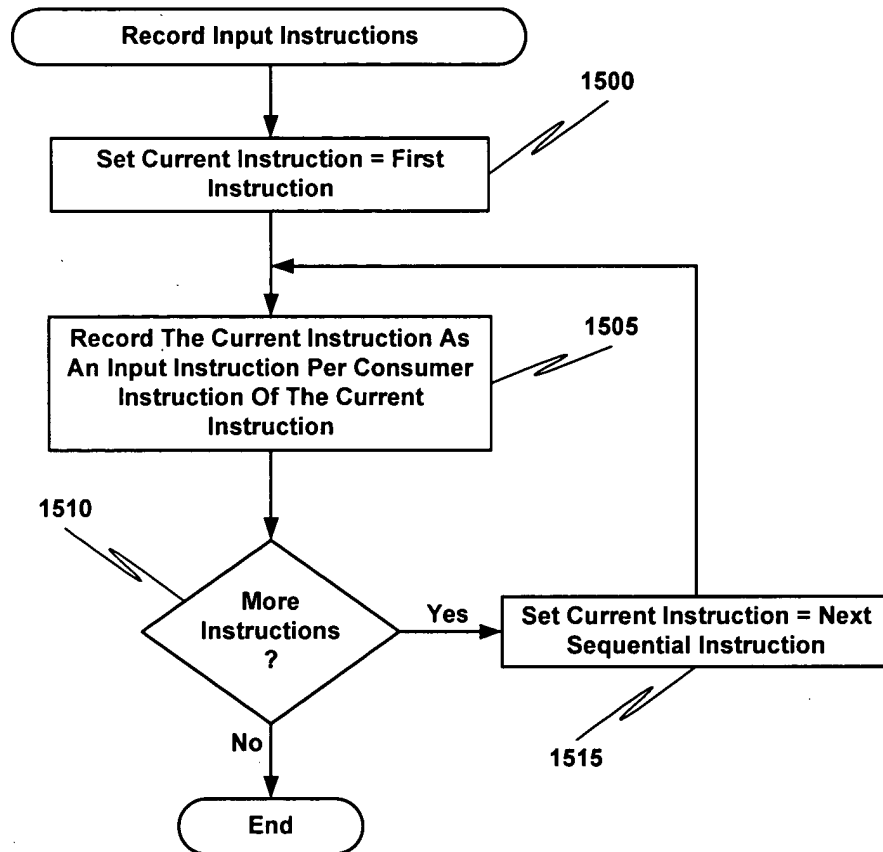


FIG. 15

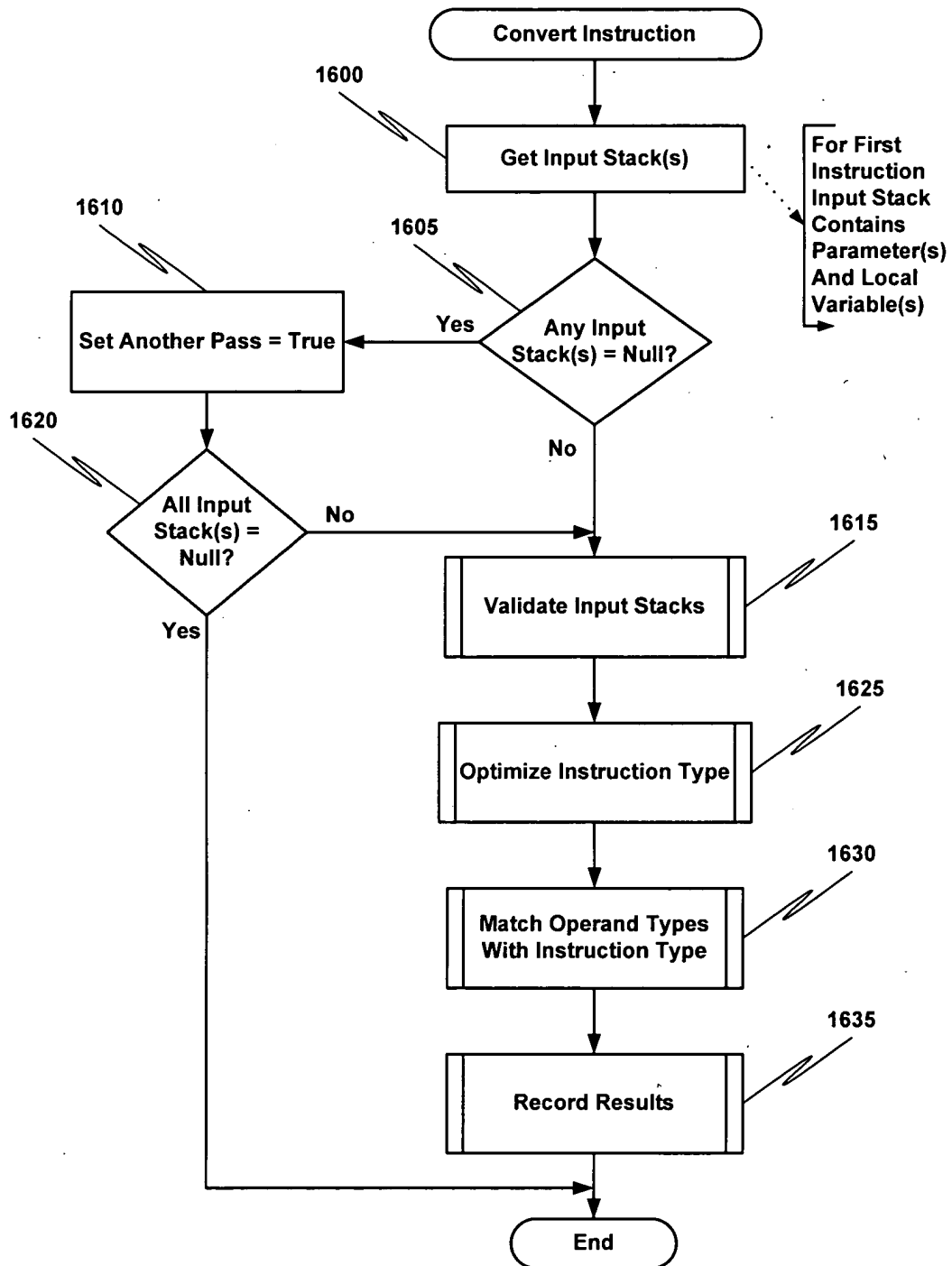


FIG. 16

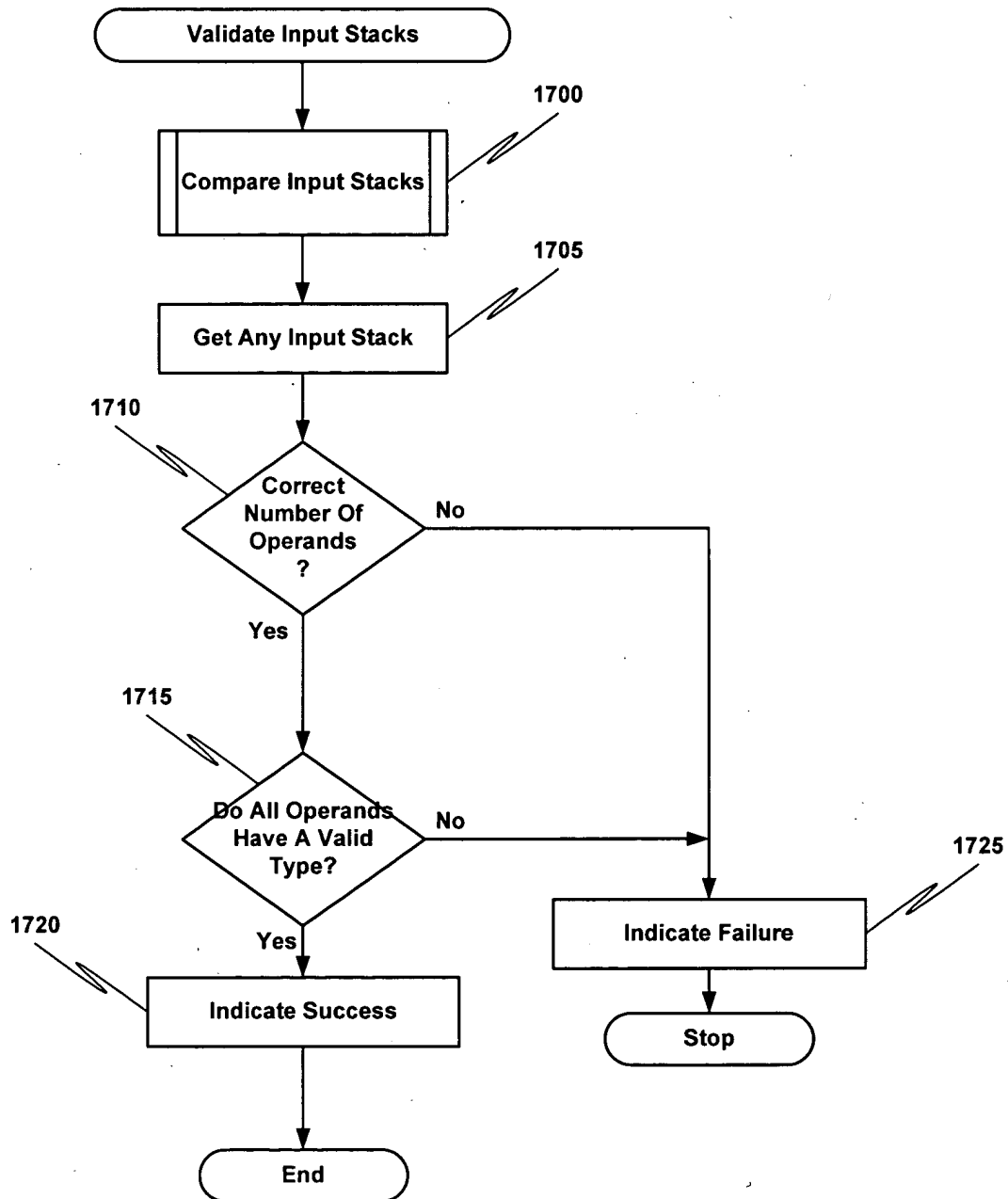


FIG. 17

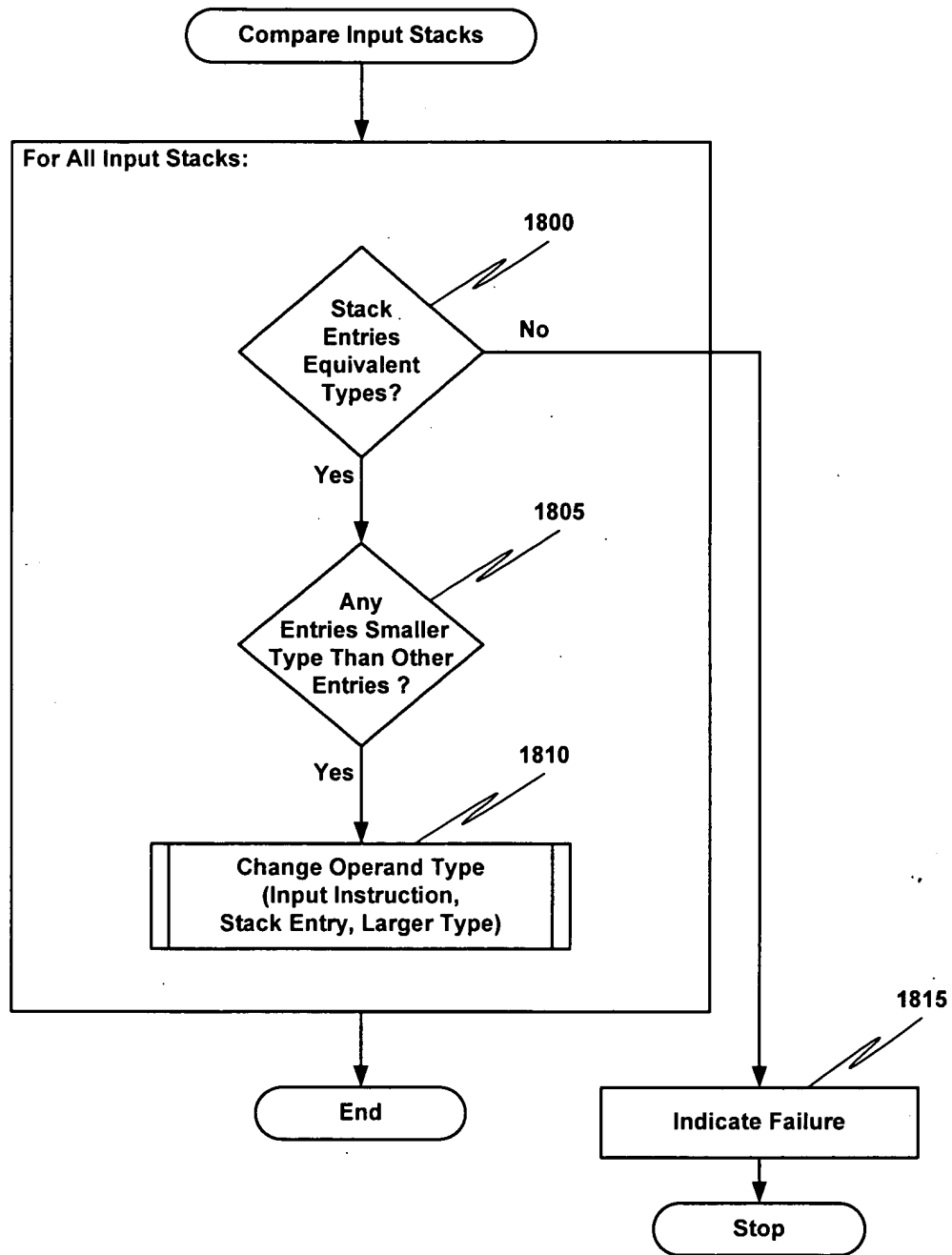


FIG. 18

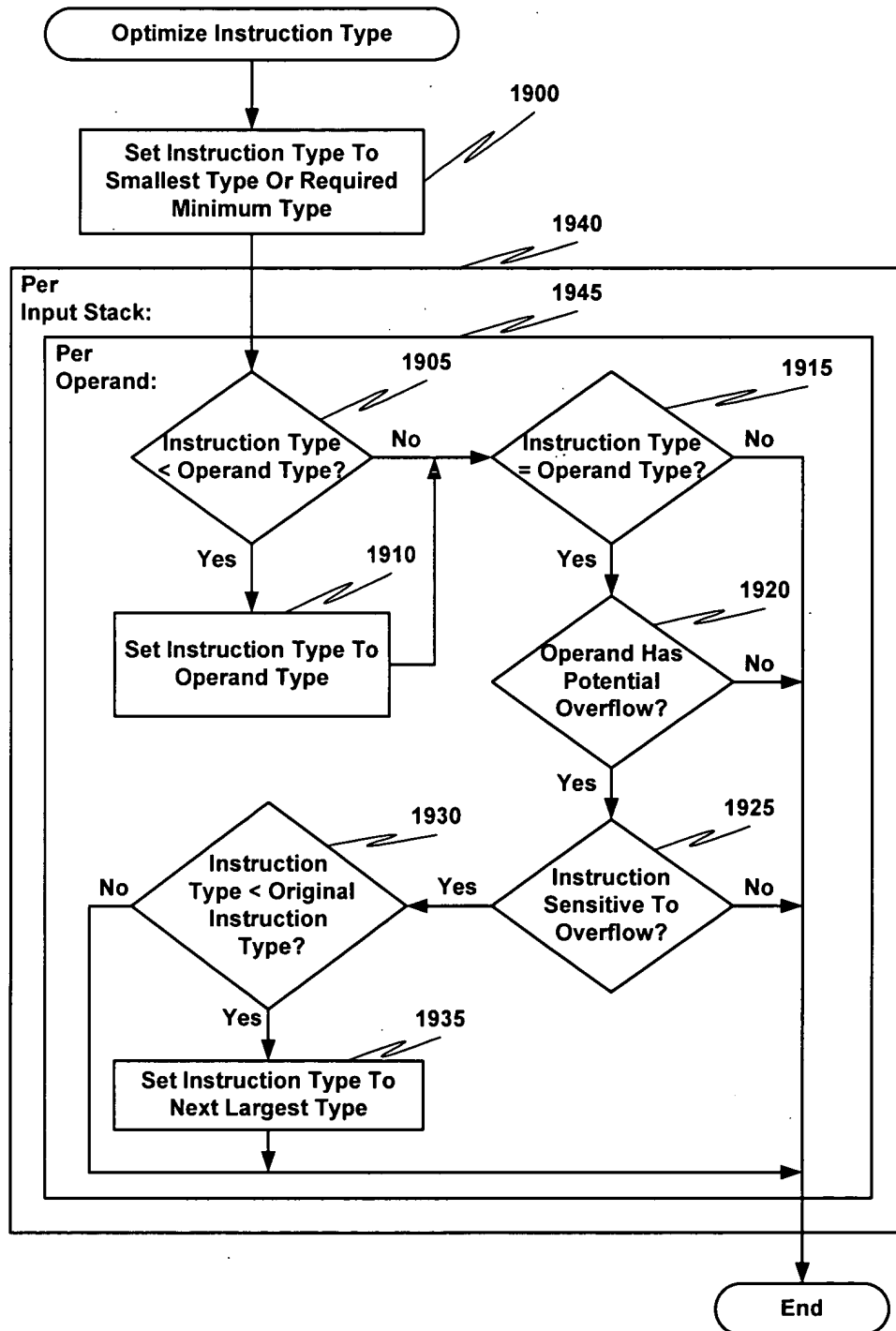


FIG. 19

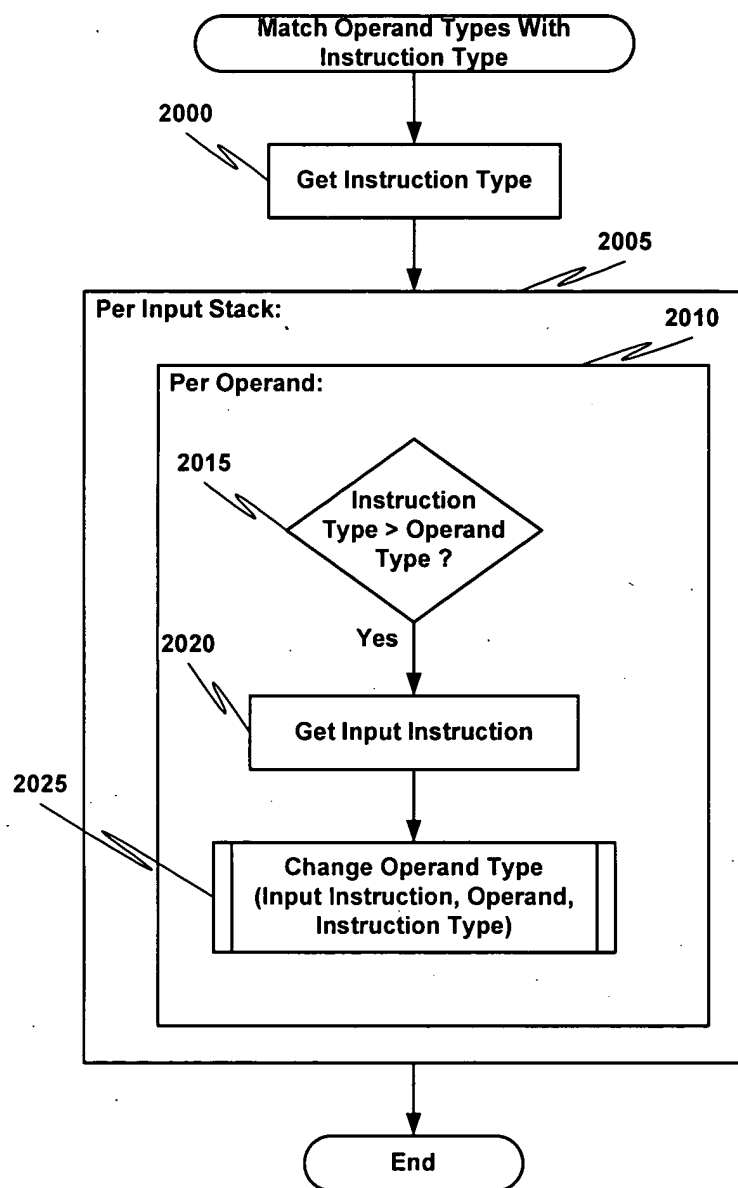


FIG. 20

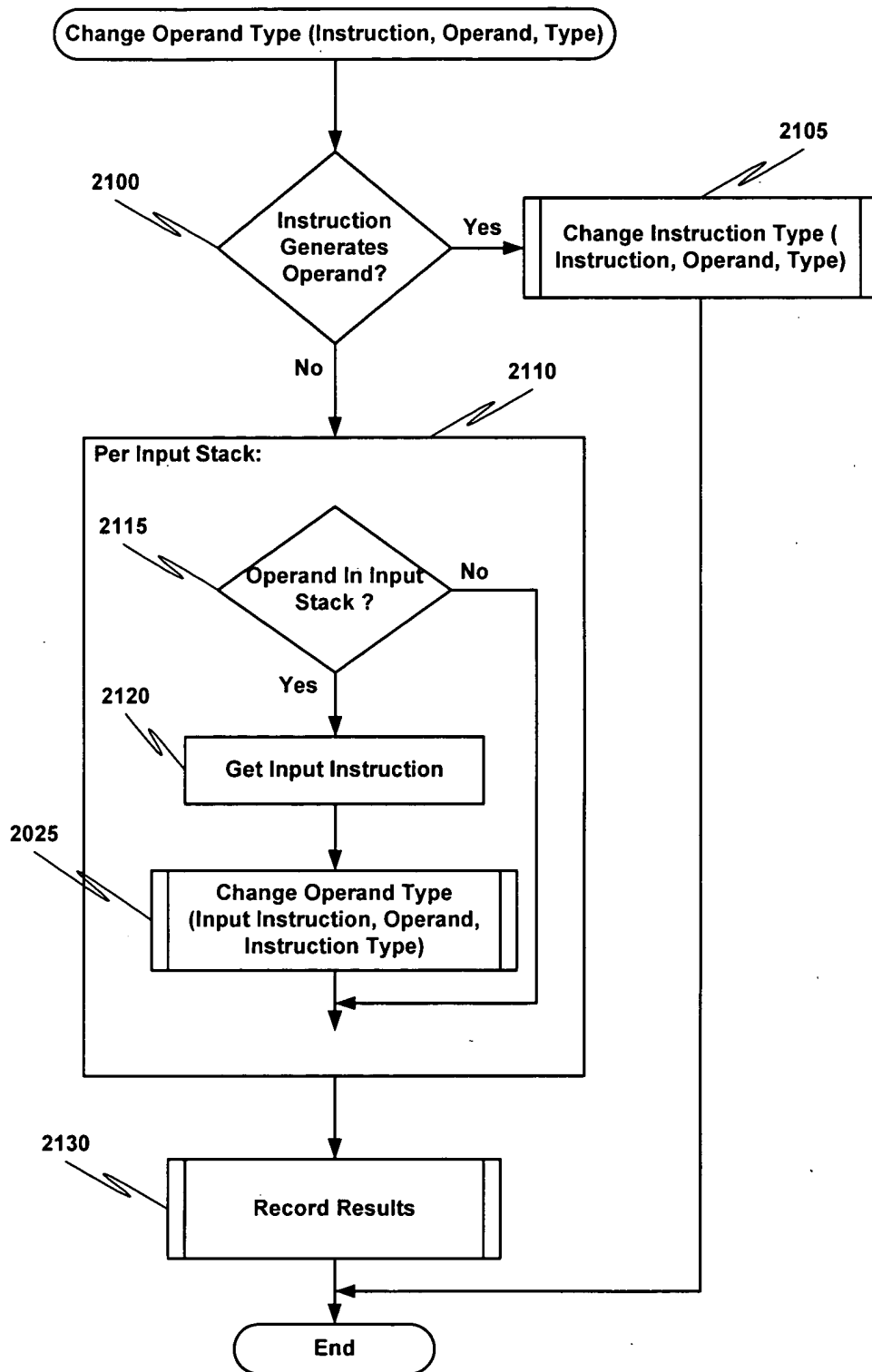


FIG. 21

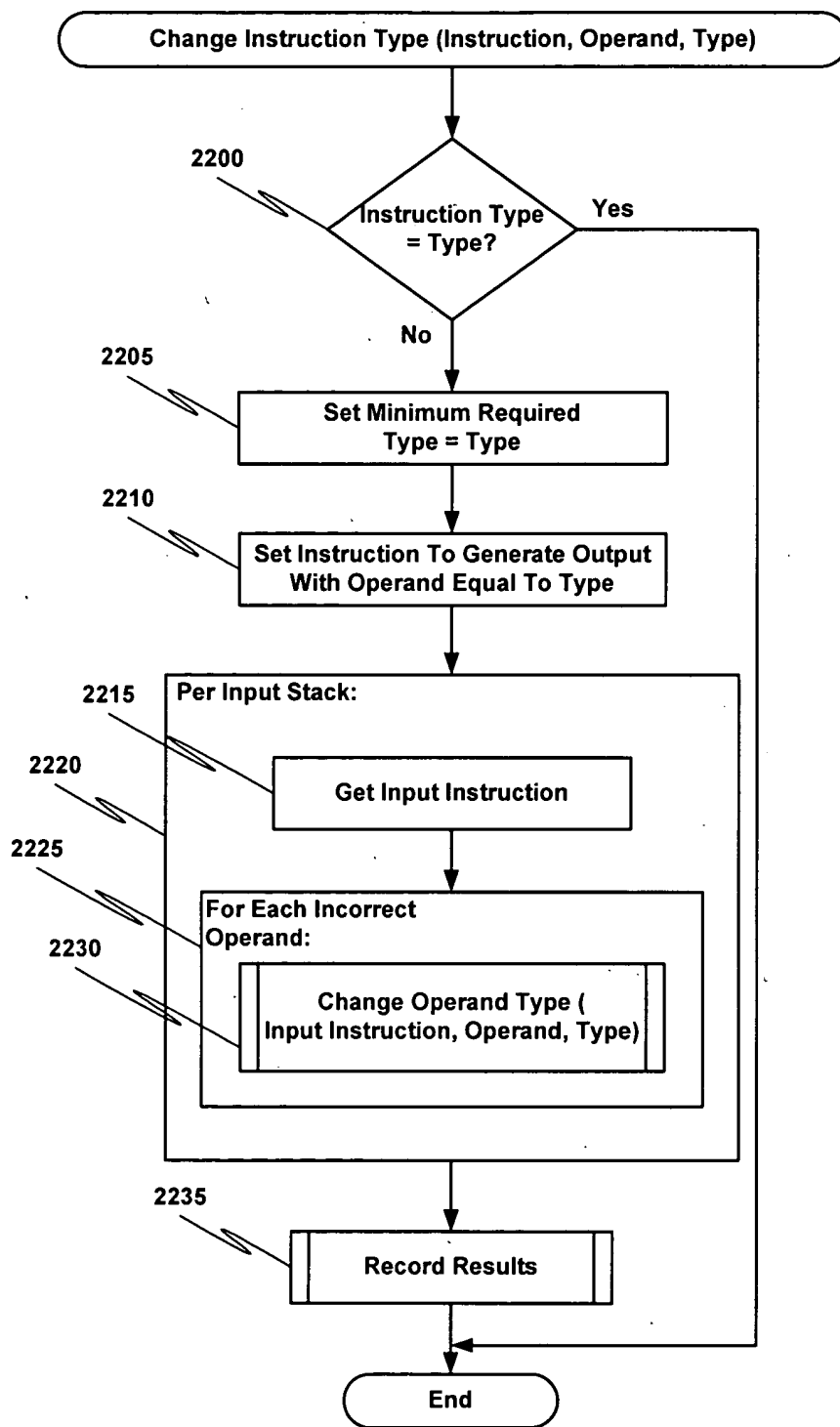


FIG. 22

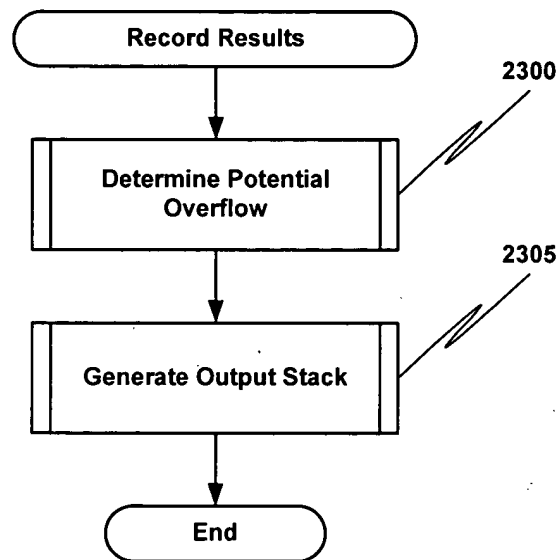


FIG. 23

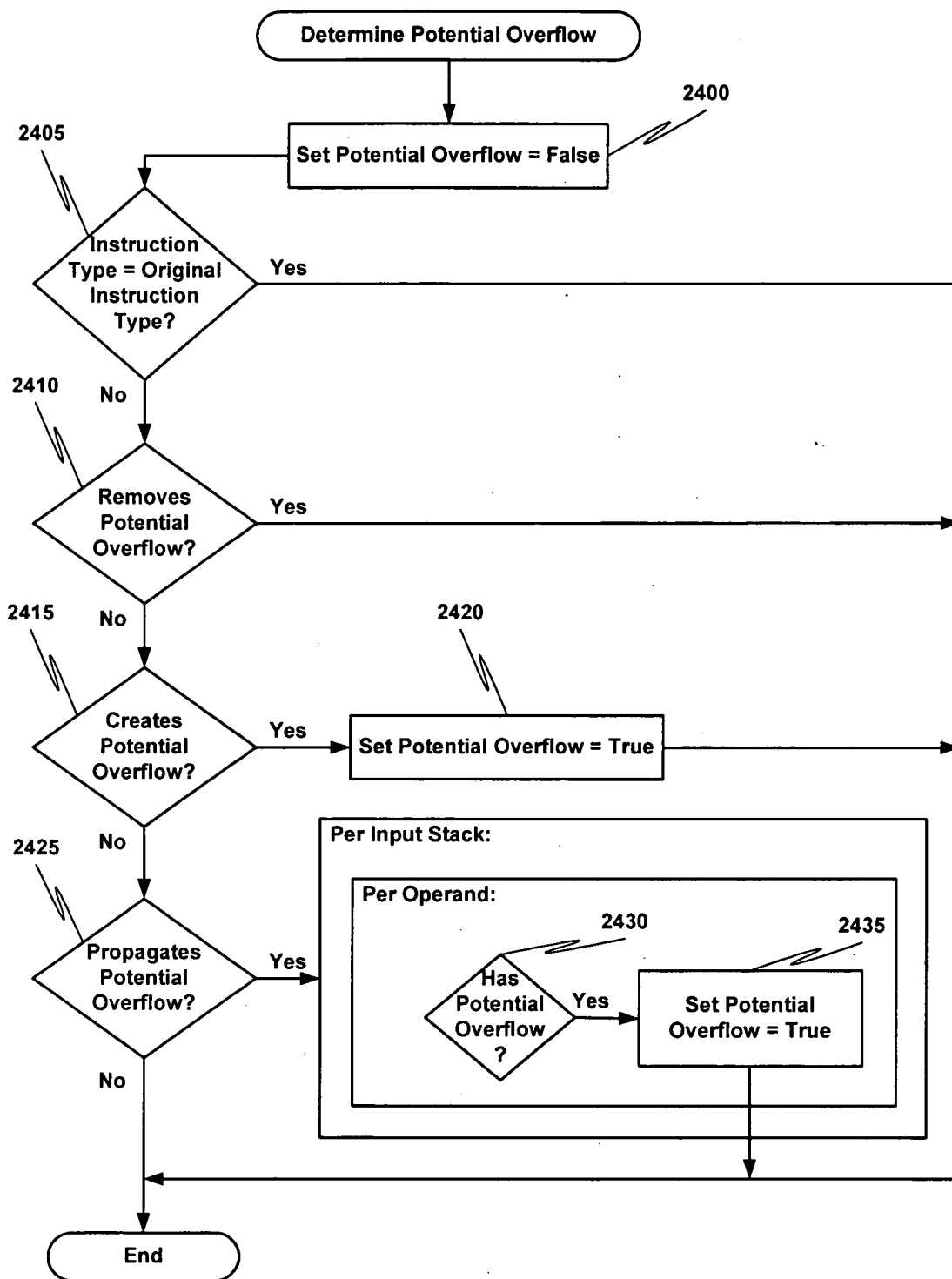


FIG. 24A

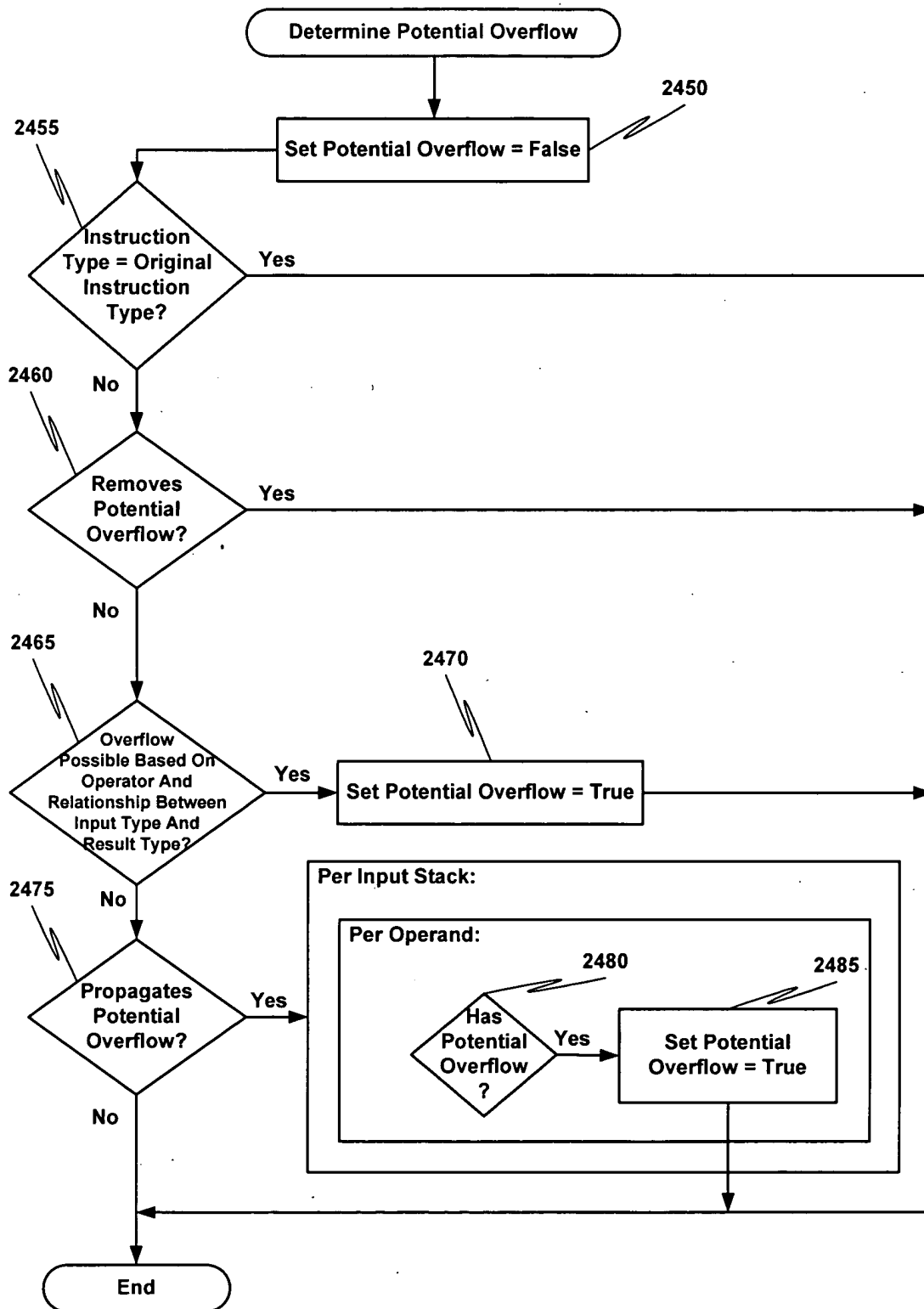


FIG. 24B

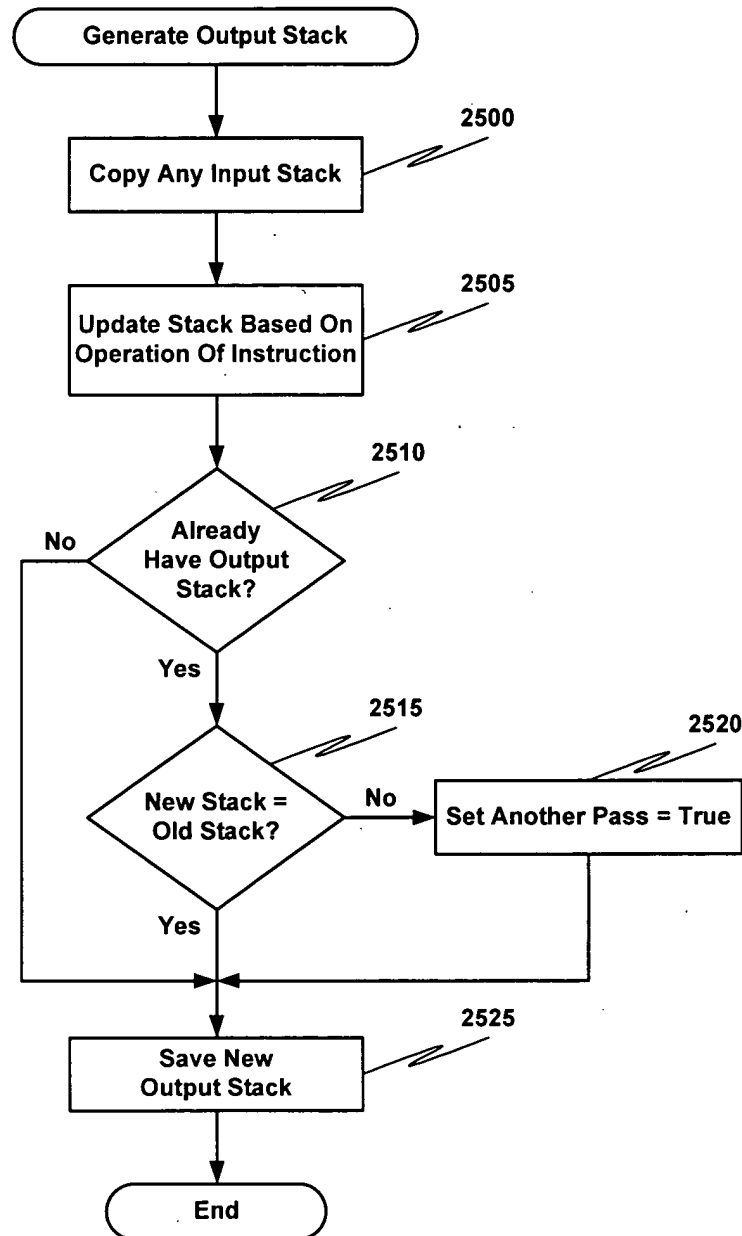


FIG. 25

Preconditions: Variables "a", "b" and "c" are declared as type "short"

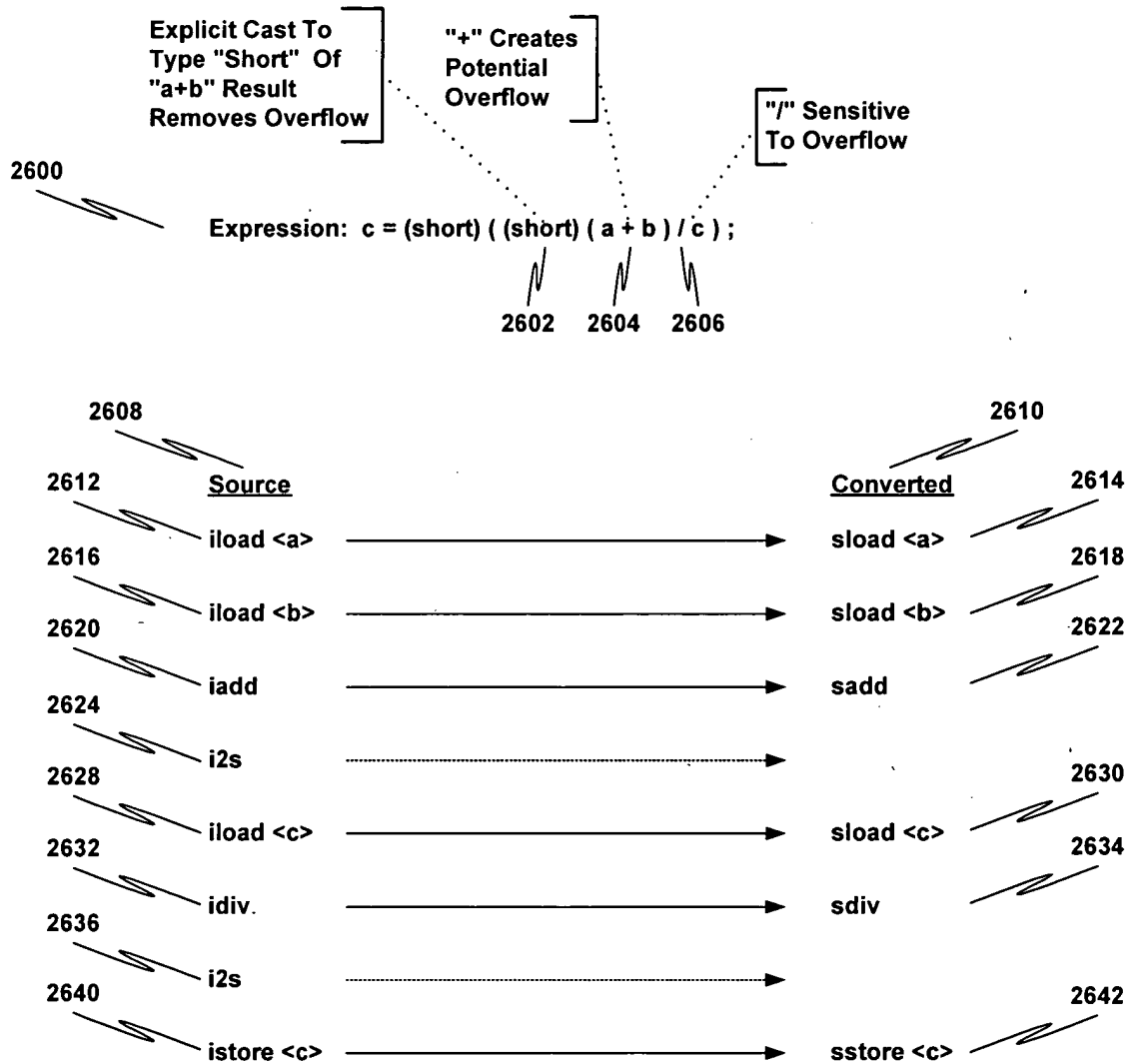


FIG. 26

Preconditions: Variables "a", "b" and "c" are declared as type "short"

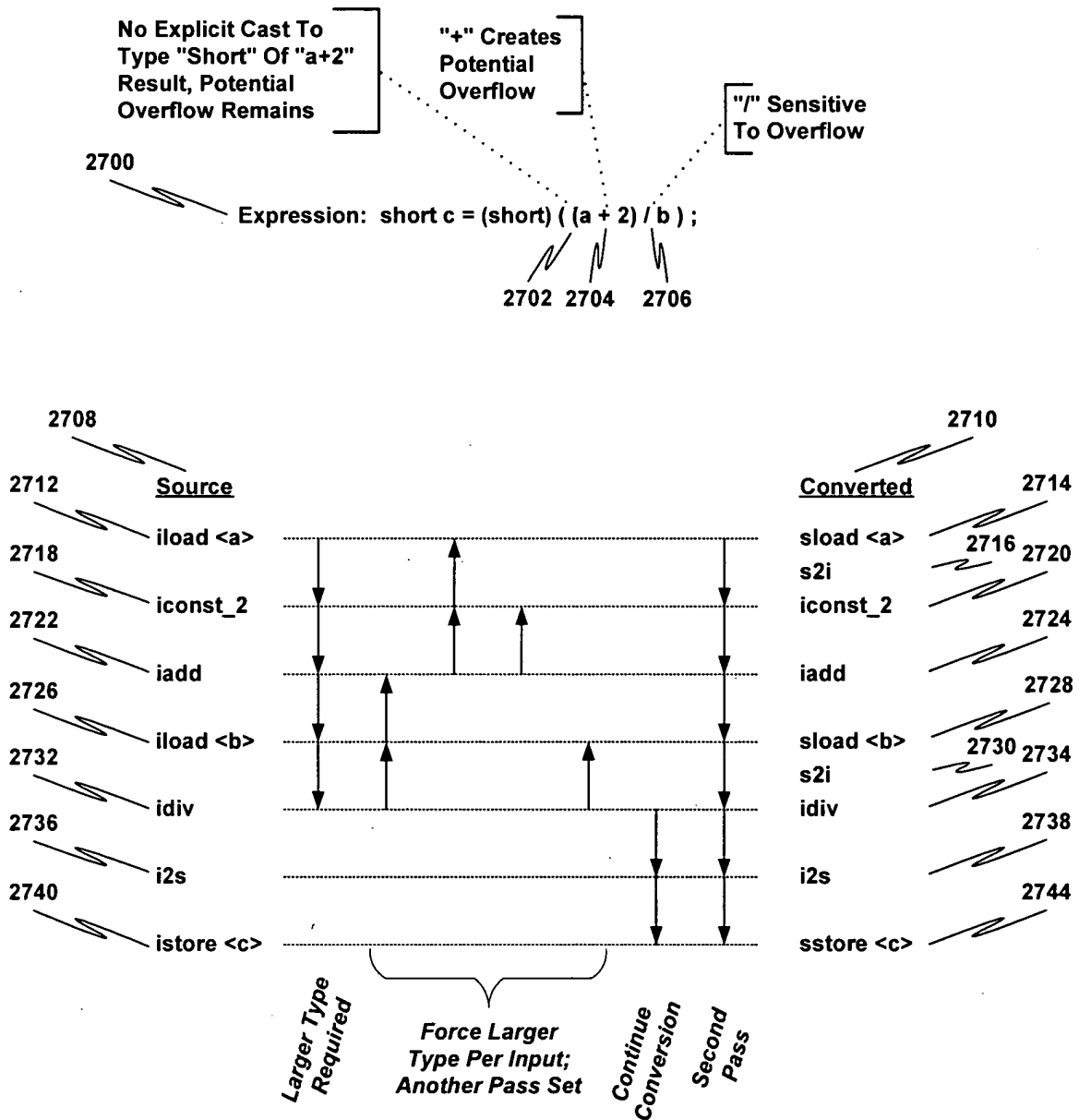


FIG. 27

Preconditions: Variables "a", "b" and "c" are declared as type "short"

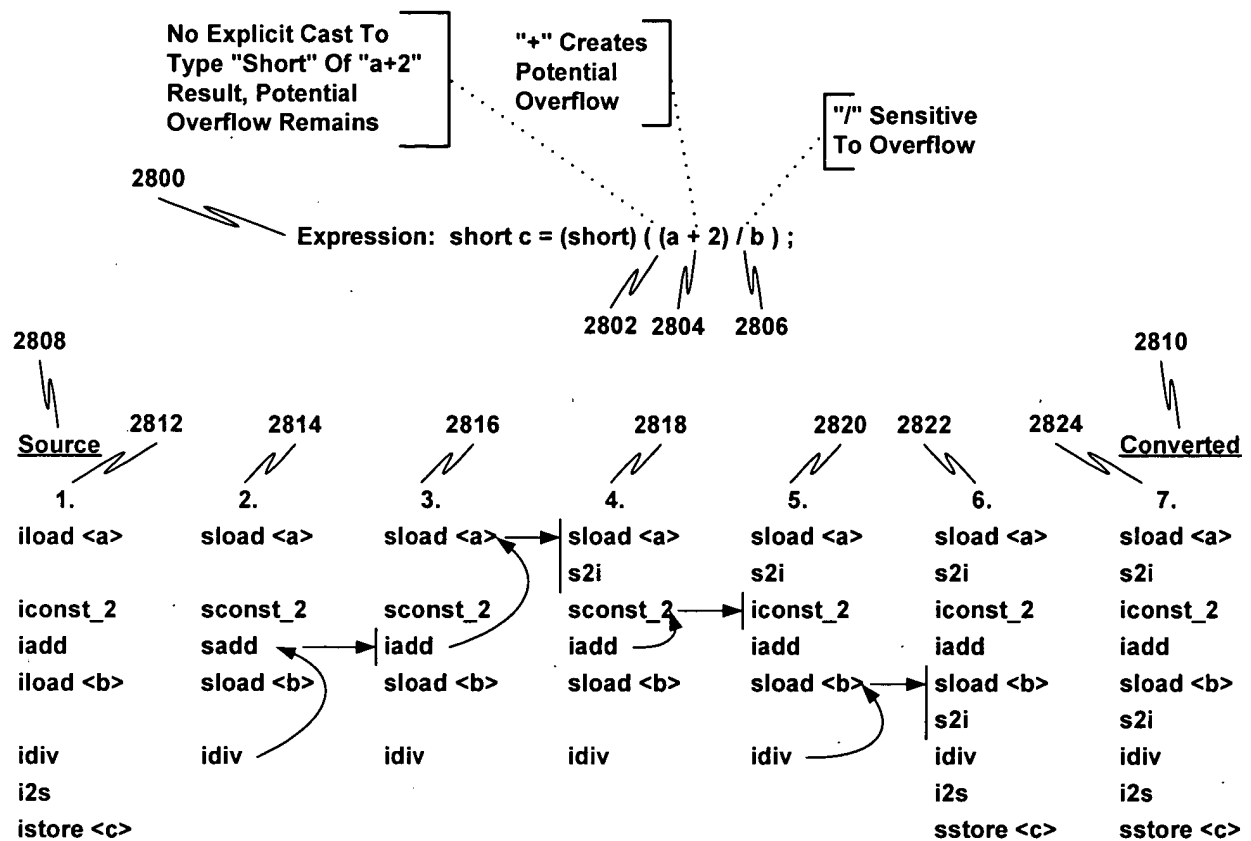


FIG. 28

Preconditions:

Variables "x", and "b" are declared as type "short"
 Variable "a", "c", and "d" are declared as type "int"

Source code:

```

if ( x != 0 )
    d = a + b;
else
    d = a + c;
  
```

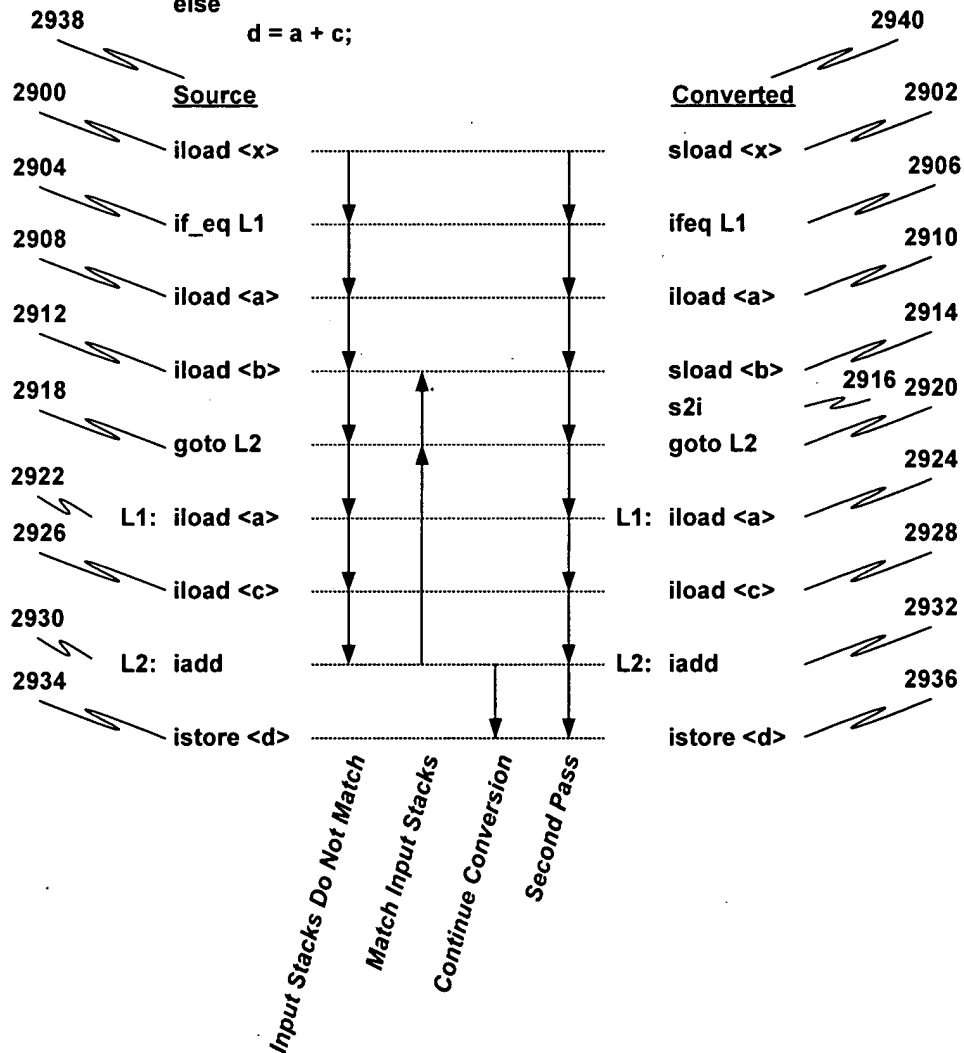


FIG. 29

Preconditions:

Variables "x", and "b" are declared as type "short"

Variable "a", "c", and "d" are declared as type "int"

Source code:

```
if ( x != 0 )  
    d = a + b;  
else  
    d = a + c;
```

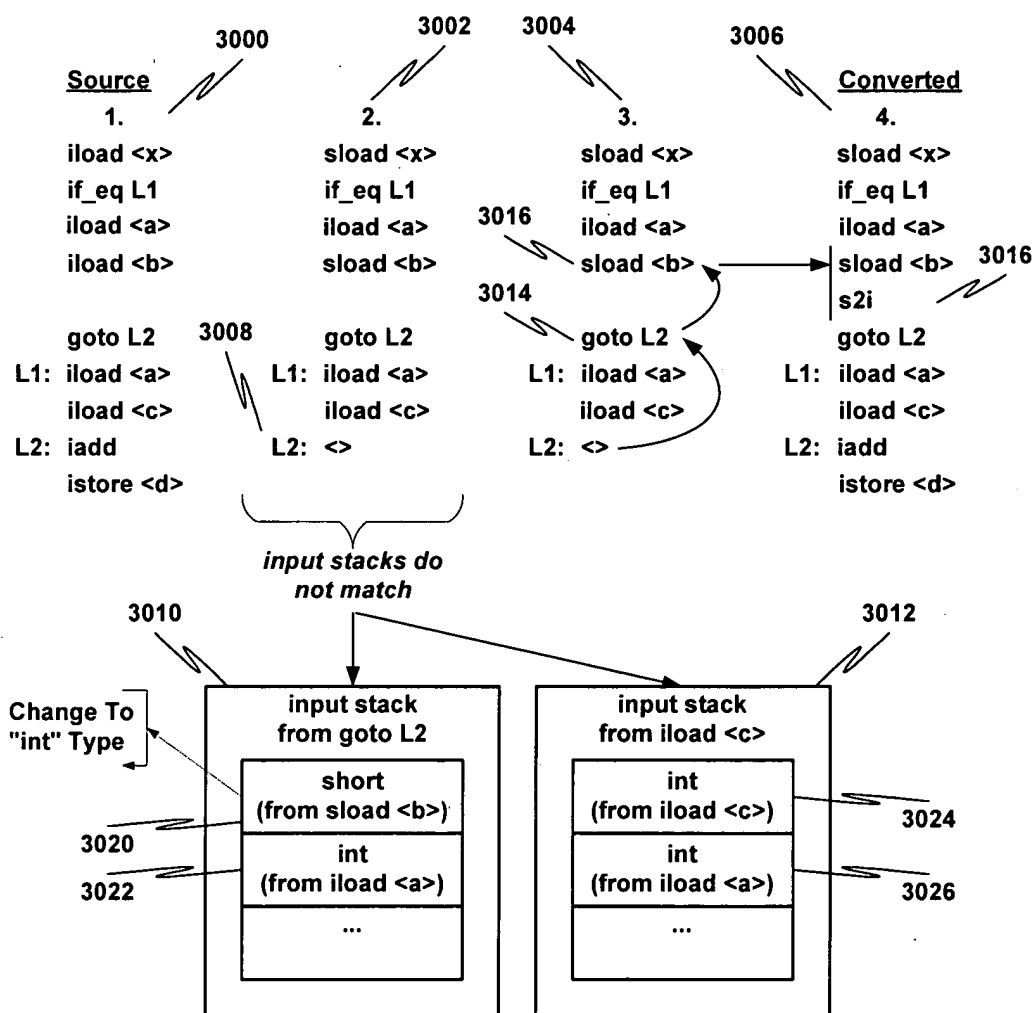


FIG. 30

Preconditions:

Variables "a", "b" and "c" are declared as type "short"
Variable "d" is declared as type "int"

Source Code:

c = (short) (a * b);
d = b + d;

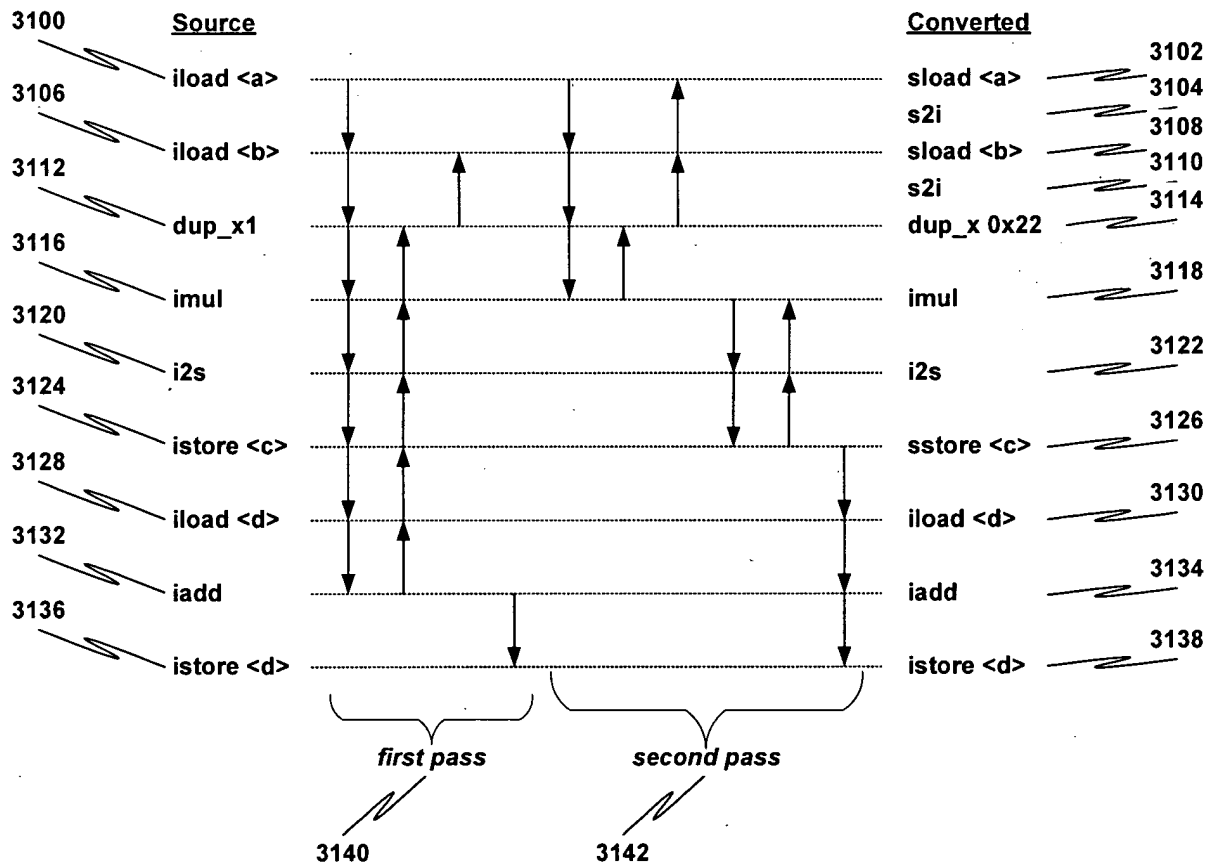


FIG. 31

Preconditions:
Variables "a", "b" and "c" are declared as type "short"
Variable "d" is declared as an type "int"

Source Code:
c = (short) (a * b);
d = b + d;

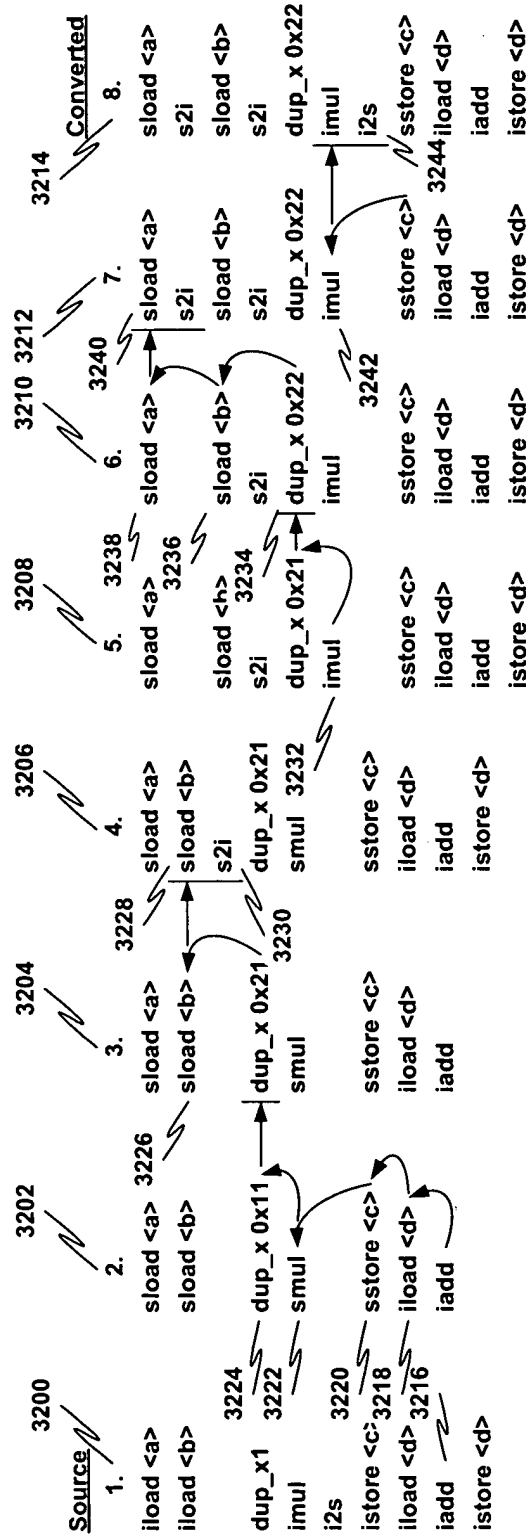


FIG. 32